IN THE CLAIMS

1. (Currently Amended) A system for communication between object request brokers (ORB), comprising:

a non-CORBA object request broker executing on a ~~first~~ client system and providing inter-object communication support between the ~~first~~ client system and a ~~second~~ server system, the ~~first~~ client system connected to the ~~second~~ server system by a network, the non-CORBA object request broker operable to generate a class with a type code and a communication protocol without generating a stub or a skeleton associated with Common Object Request Broker Architecture (CORBA) object request brokers; and

a reference object in the non-CORBA object request broker operable to encode outgoing communications into an Internet Inter-ORB Protocol (IIOP) format according to the communication protocol in the generated class, the reference object further operable to decode incoming communications from the Internet Inter-ORB Protocol (IIOP) format into a format native to the non-CORBA object request broker.

2. (Currently Amended) The system of Claim 1, further comprising a CORBA object request broker executing on the ~~second~~ server system.

3. (Currently Amended) The system of Claim 1, further comprising one or more streamers coupled to the reference object, the one or more streamers corresponding in number to methods of a target object, the one or more streamers serially sending bytes of outgoing communications to the ~~second~~ server system.

4. (Currently Amended) The system of Claim 1, further comprising a client application on the ~~first~~ client system.

5. (Currently Amended) The system of Claim 1, further comprising a target object on the ~~second~~ server system.

6. (Previously Presented) The system of Claim 1, wherein the class is generated from Interface Description Language (IDL) definitions.

7. (Previously Presented) The system of Claim 6, wherein the non-CORBA object request broker provides an ORB-specific implementation of the IDL class having information to communicate with other ORBs.

8. (Original) The system of Claim 1, wherein a remote proxy sends the outgoing communication to the reference object.

9. (Currently Amended) The system of Claim 8, wherein the remote proxy receives the outgoing communication from an application on the ~~first~~ client system.

10. (Currently Amended) The system of Claim 1, wherein the reference object receives incoming communications from the ~~second~~ server system.

11. (Previously Presented) The system of Claim 1, wherein the type code identifies a structure corresponding to an Interface Description Language (IDL) definition and provides communication support between CORBA and non-CORBA ORBs.

12. (Currently Amended) A method for communication between object request brokers (ORB), comprising:

invoking a method of a target object on a ~~first~~ server system by an application on a ~~second~~ client system;

generating on the client system a class with a type code and a communication protocol without generating a stub or a skeleton associated with Common Object Request Broker Architecture (CORBA) compliant object request brokers;

forwarding the method invocation to a reference object associated with the communication protocol in a ~~second~~ client object request broker executing on the ~~second~~ client system;

encoding the method invocation into Internet Inter-ORB Protocol (IIOP) format;

sending the encoded method invocation to a ~~first~~ server object request broker executing on the ~~first~~ server system; and

invoking the method on the target object.

13. (Currently Amended) The method of Claim 12, wherein sending the encoded method invocation includes:

forwarding the encoded method invocation to one of one or more streamer objects corresponding to a method invoked by the encoded method invocation; and

serially streaming bytes of the encoded method invocation to the ~~first~~ server object request broker.

14. (Currently Amended) The method of Claim 12, further comprising:

forwarding a result of the method invocation to the ~~first~~ server object request broker;

transmitting the result to the ~~second~~ client object request broker executing on the ~~second~~ client system;

receiving the result encoded in Internet Inter-ORB Protocol (IIOP) format in the reference object;

decoding the result into a format native to the ~~second~~ client object request broker; and

forwarding the result to the application.